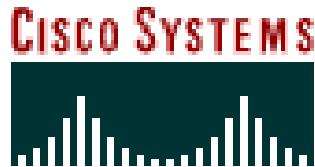


# Stream Control Transmission Protocol - SCTP



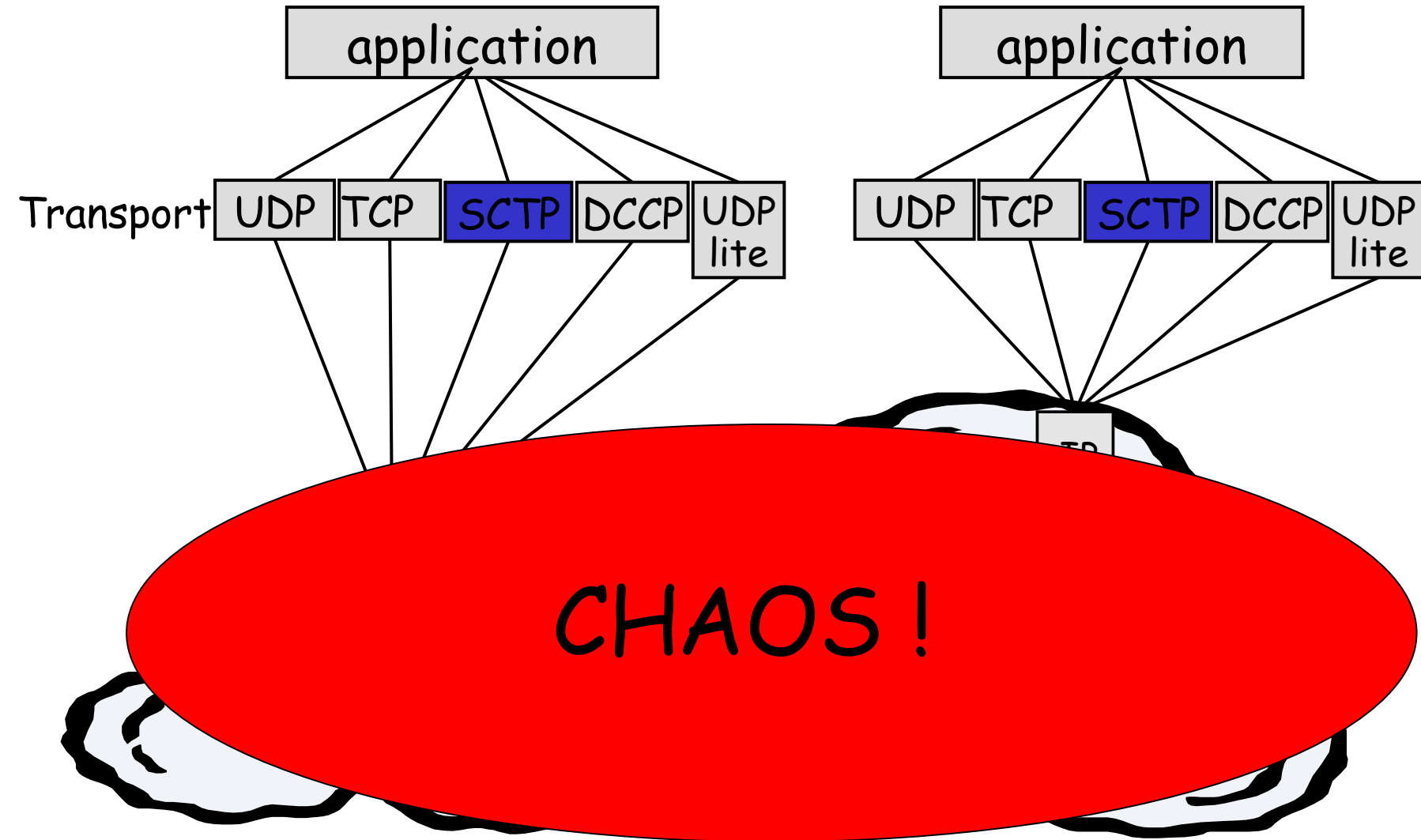
Protocol · Engineering · Laboratory

**Randall Stewart**  
**NSSTG**

**Prof. Paul Amer**  
**Computer Science Dept**

Located at <http://sctp.org> Under the RFC tab

# Where does it fit in?



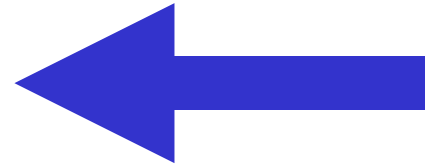
# What is SCTP?

## *Start with TCP:*

- reliable (retransmissions)
- congestion-controlled
- flow-controlled
- connection-oriented
- selective acknowledgments

## *Add:*

- “association” 4-way handshake
  - to reduce vulnerability to DOS attacks
- framing, unordered service
  - preserve message boundaries
- multistreaming
  - not one ordered stream, but 64K independent ordered streams
- multihoming
  - not one, but a set of IP addresses per endpoint
- reachability
  - heartbeating keeps track of endpoint status

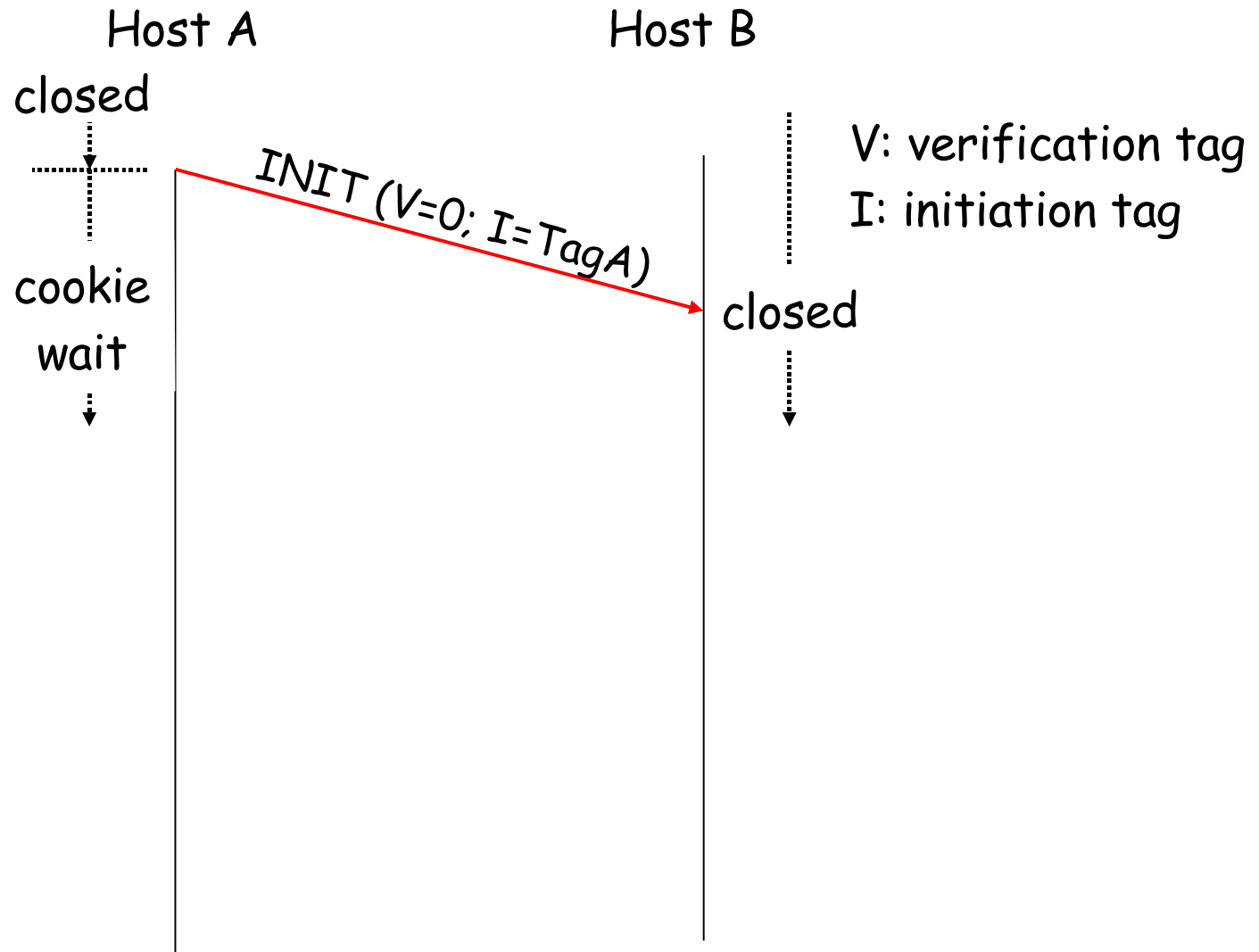


# SCTP Overview

<b>Services/Features</b>	<b>SCTP</b>	<b>TCP</b>	<b>UDP</b>
Connection-oriented	yes	yes	no
Full duplex	yes	yes	yes
Reliable data transfer	yes	yes	no
Partial-reliable data transfer	optional	no	no
Flow control	yes	yes	no
TCP-friendly congestion control	yes	yes	no
ECN capable	yes	yes	no
Ordered data delivery	yes	yes	no
Unordered data delivery	yes	no	yes
Uses selective ACKs	yes	optional	no
Path MTU discovery	yes	yes	no
Application PDU fragmentation	yes	yes	no
Application PDU bundling	yes	yes	no
Preserves application PDU boundaries	yes	no	yes
Multistreaming	yes	no	no
Multihoming	yes	no	no
Protection against SYN flooding attack	yes	no	n/a
Allows half-closed connections	no	yes	n/a
Reachability check	yes	yes	no
Pseudo-header for checksum	no (uses vtags)	yes	yes
Time wait state	for vtags	for 4-tuple	n/a

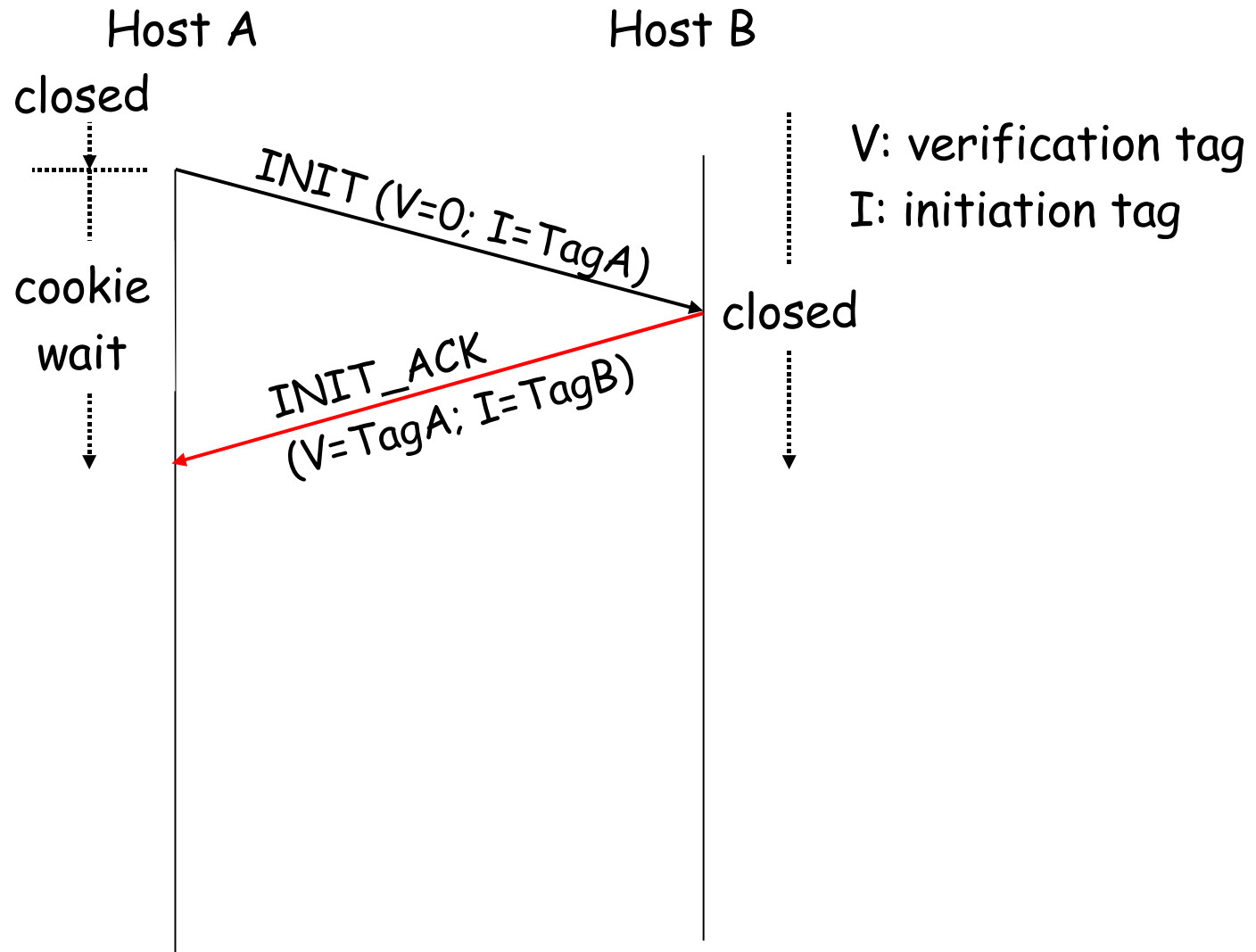
# SCTP Association setup

How many way handshake ?



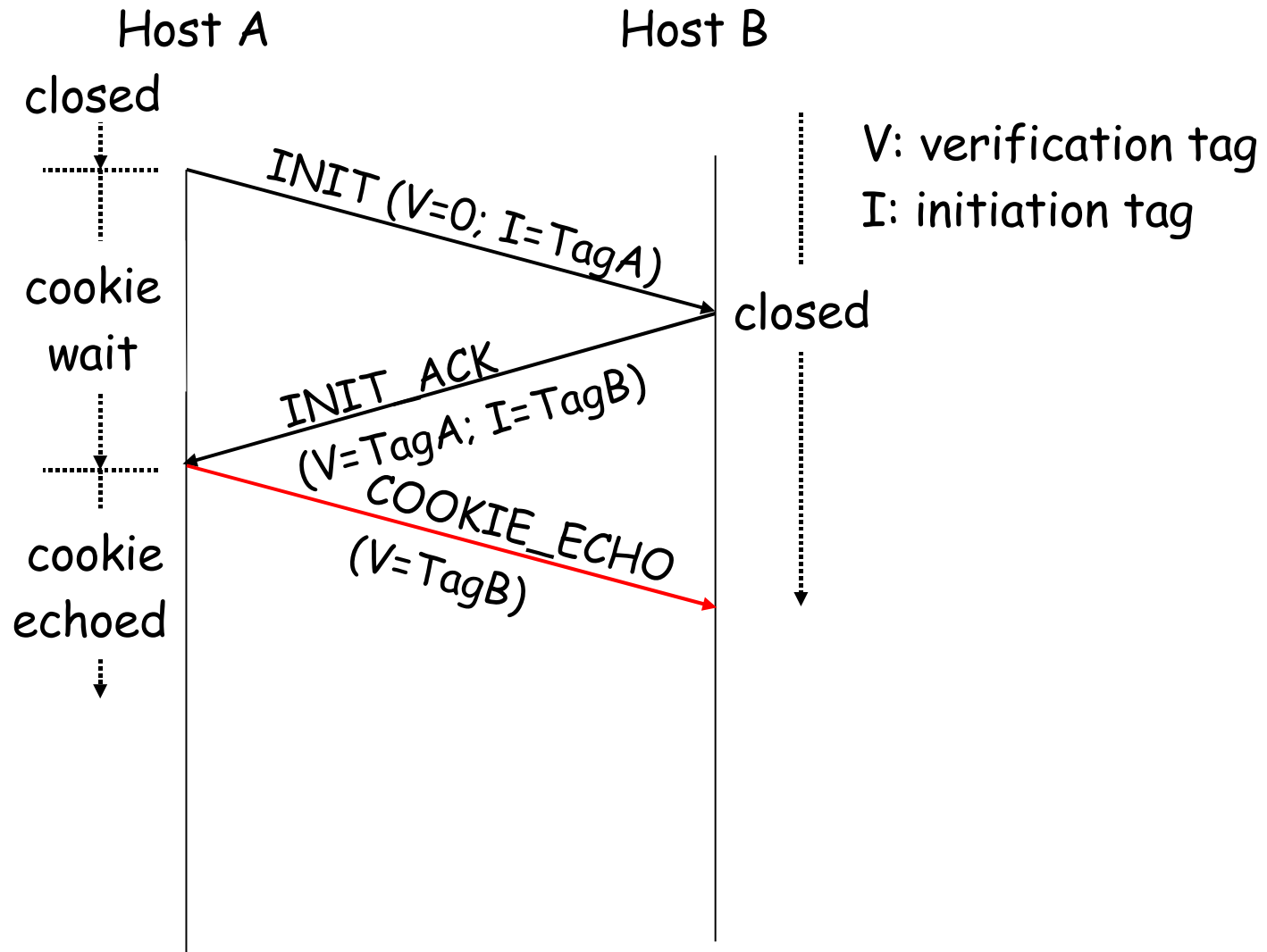
INIT PDU

# SCTP Association setup (cont'd)



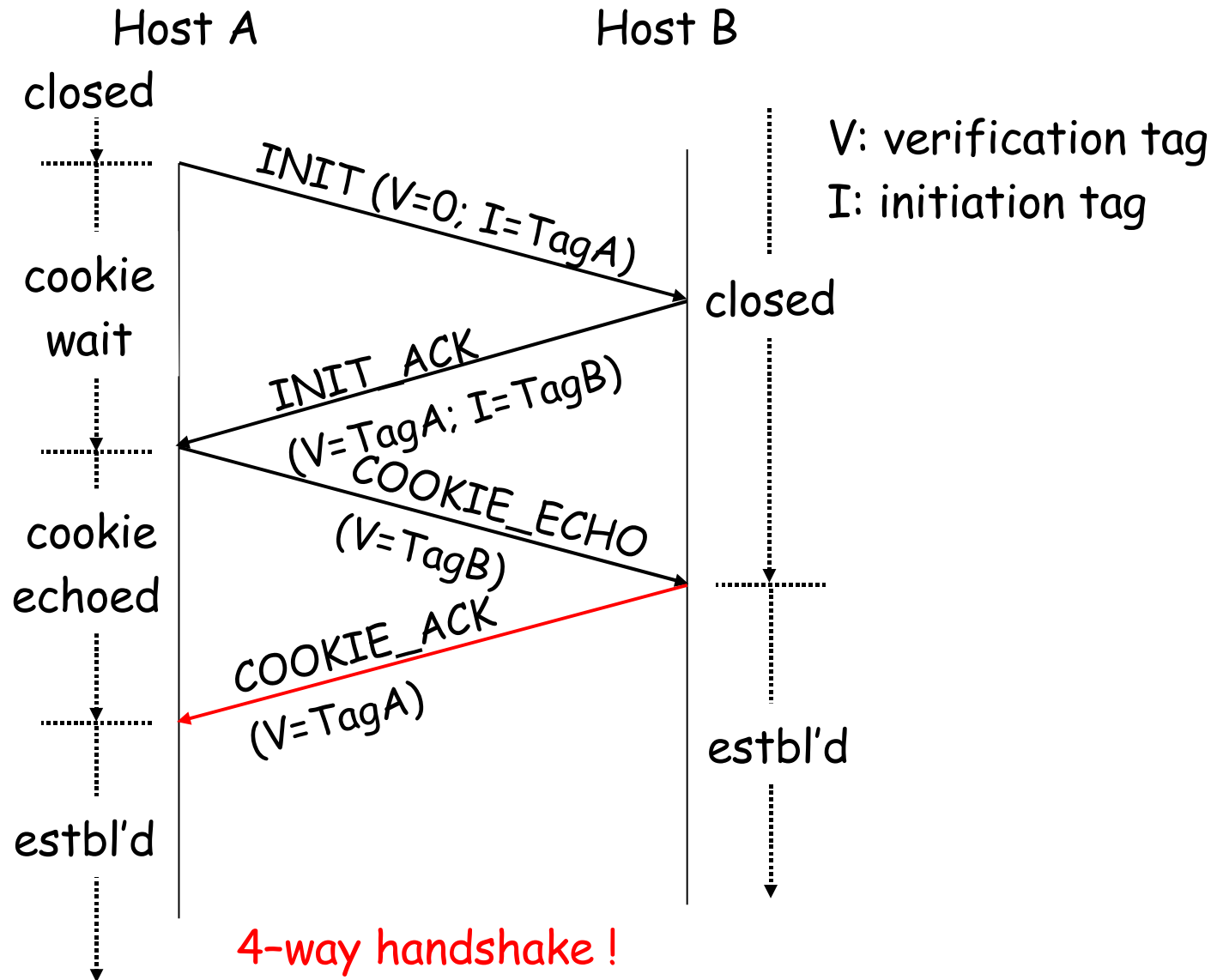
INIT ACK PDU

# SCTP Association setup (cont'd)



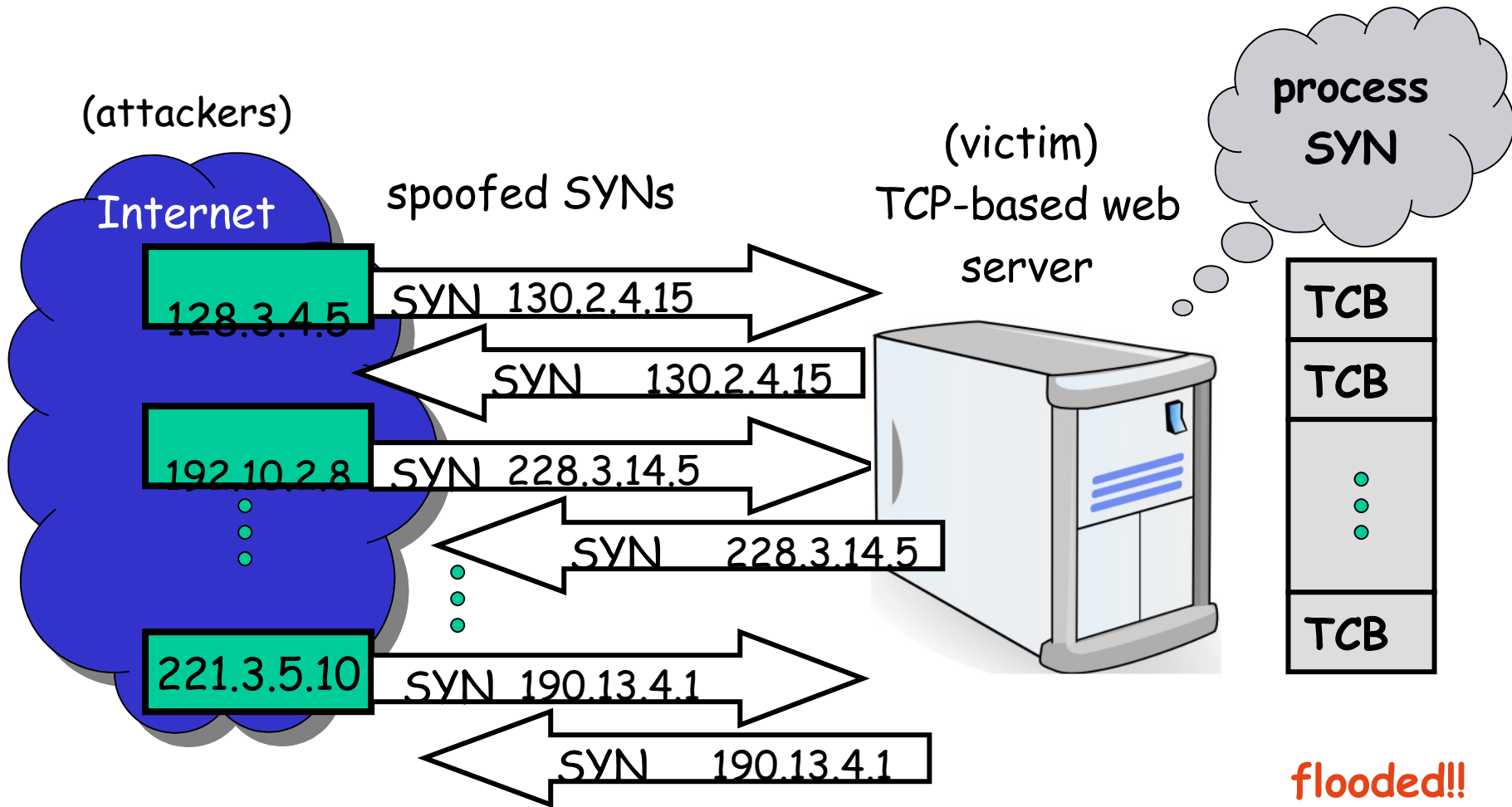
COOKIE ECHO PDU

# SCTP Association setup (cont'd)



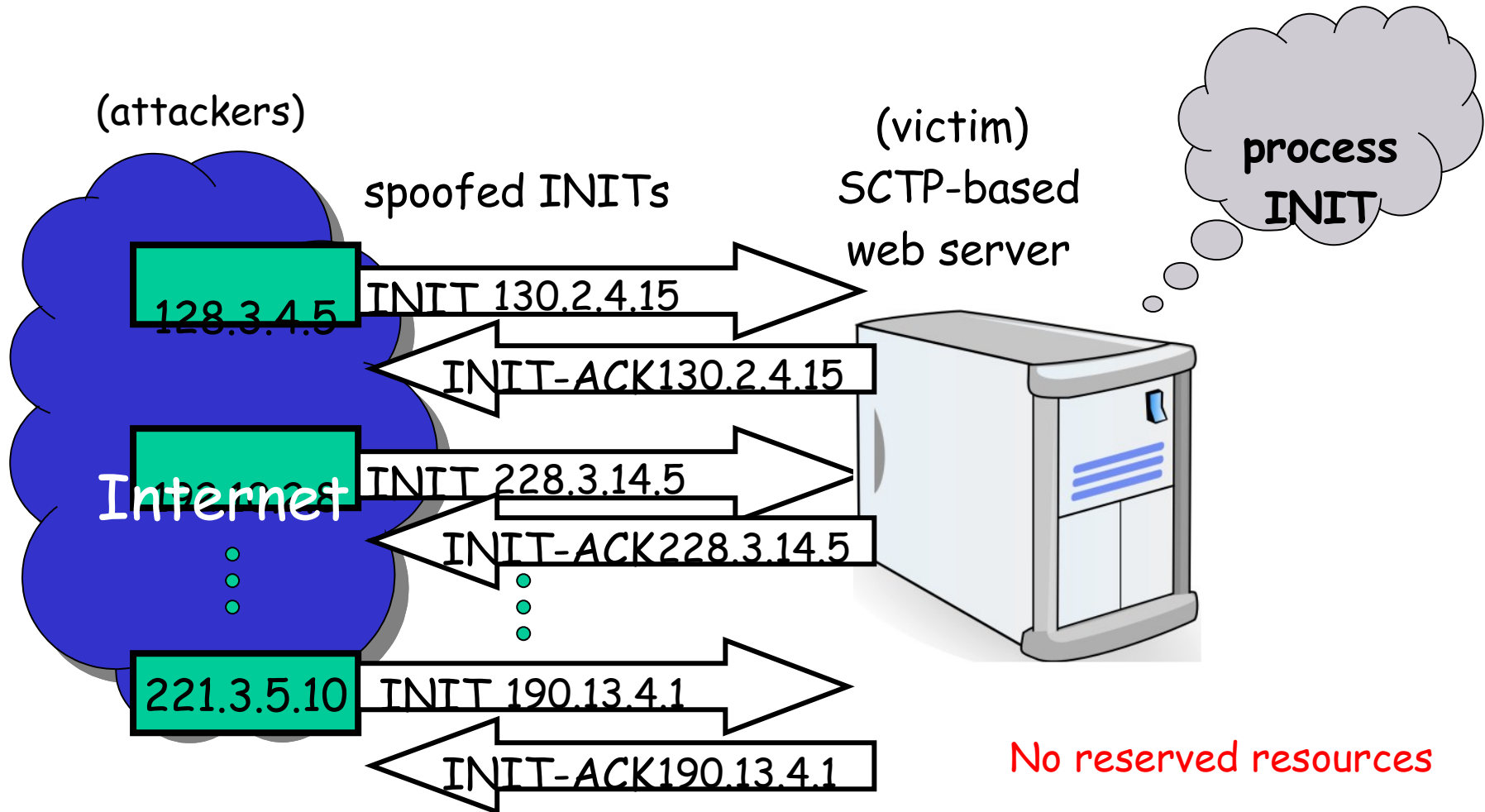
COOKIE ACK PDU

# Security: TCP Flooding Attack



TCB = Transport Control Block

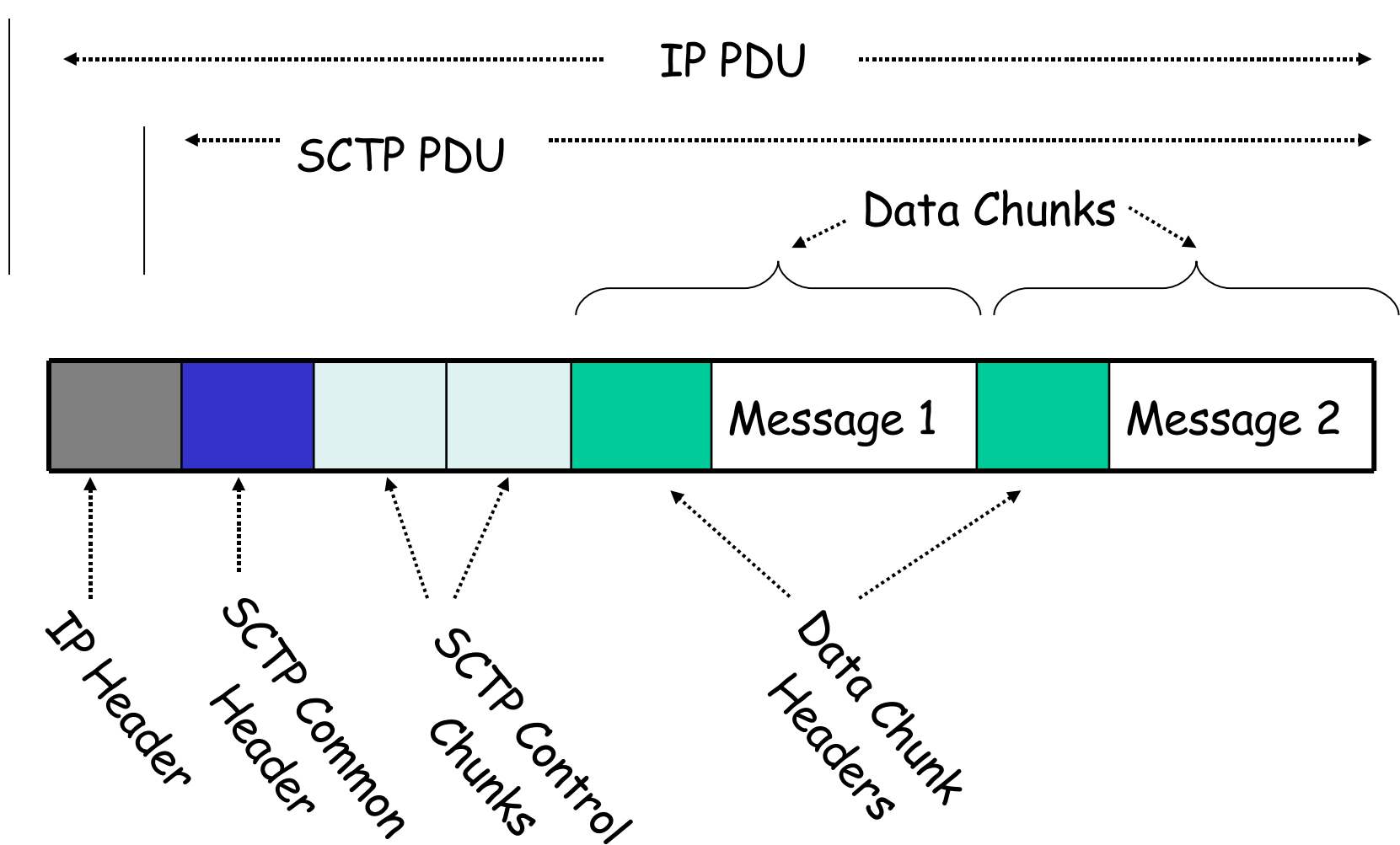
# 4-way handshake limits attack



# Message Boundaries, UDP, TCP, and SCTP

- Example - DNS
  - Response size  $\leq 512$ , UDP is used
  - If response size  $> 512$ , TCP is used
    - Application must preface the header with message length
- UDP preserves message boundaries
- TCP does not
- SCTP does
  - Data flags manage message boundaries and fragmentation

# IP Encapsulated SCTP PDU



# A Large Message Transfer

data (2760 octets)

sent from application

data to be sent

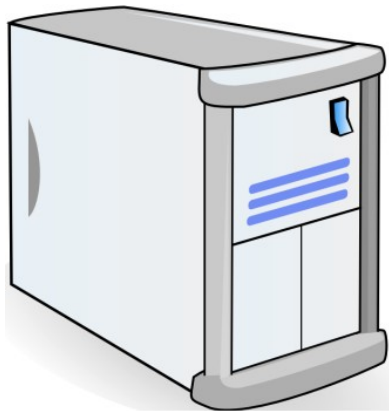
TSN	TSN	TSN	TSN	TSN	TSN
6 <sup>2</sup>	5	4	3	2	1 <sup>1</sup>

data

delivered to application

receive buffer (6)

TSN	TSN	TSN	TSN	TSN	TSN
6 <sup>2</sup>	5	4	3	2	1 <sup>1</sup>



A<sub>1</sub>

A<sub>2</sub>

TSN  
1<sup>1</sup>

TSN  
1<sup>1</sup>

B<sub>1</sub>

B<sub>2</sub>

Path MTU = 512 octets



1 - B bit set to 1

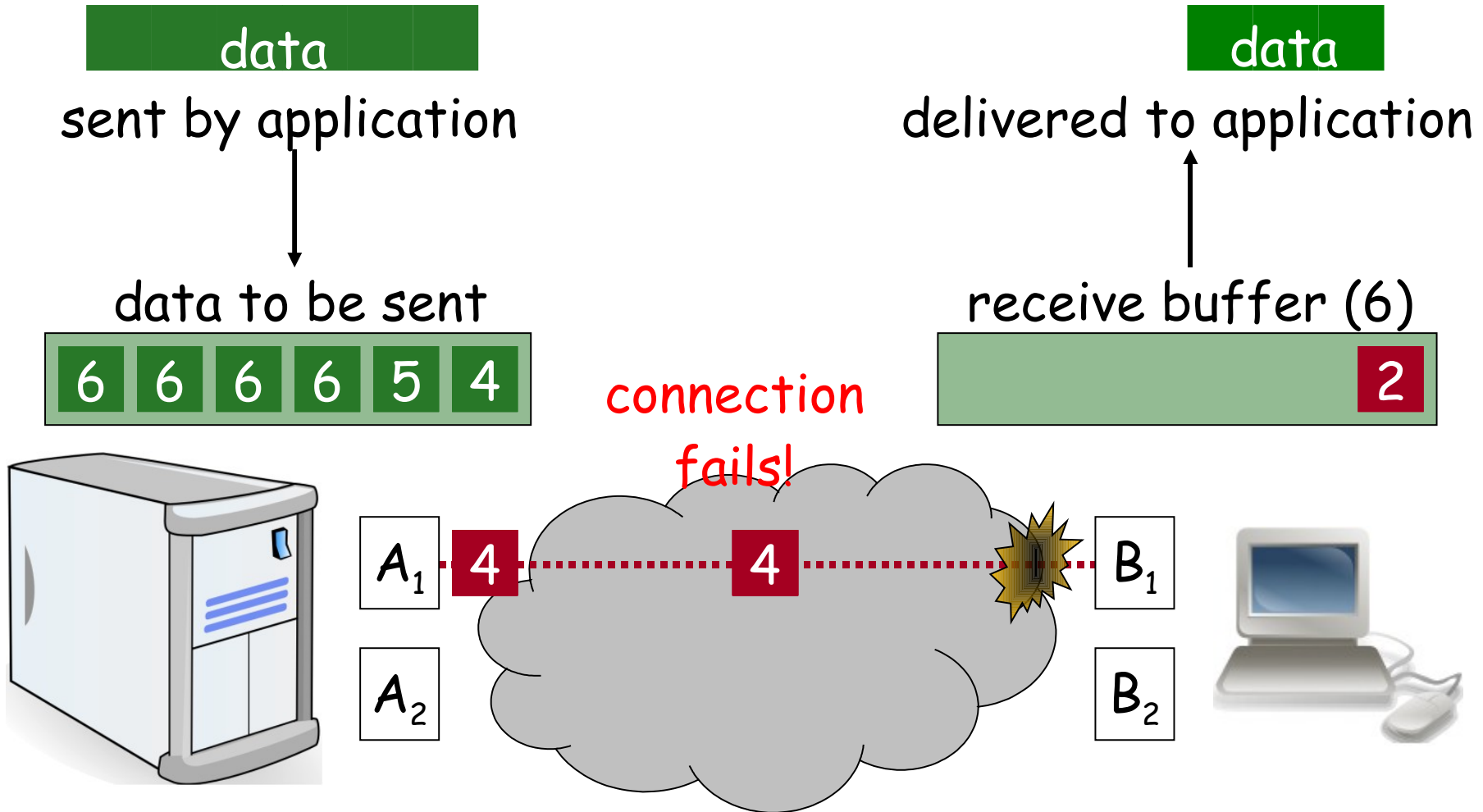
2 - E bit set to 1

# Multi-homed Considerations

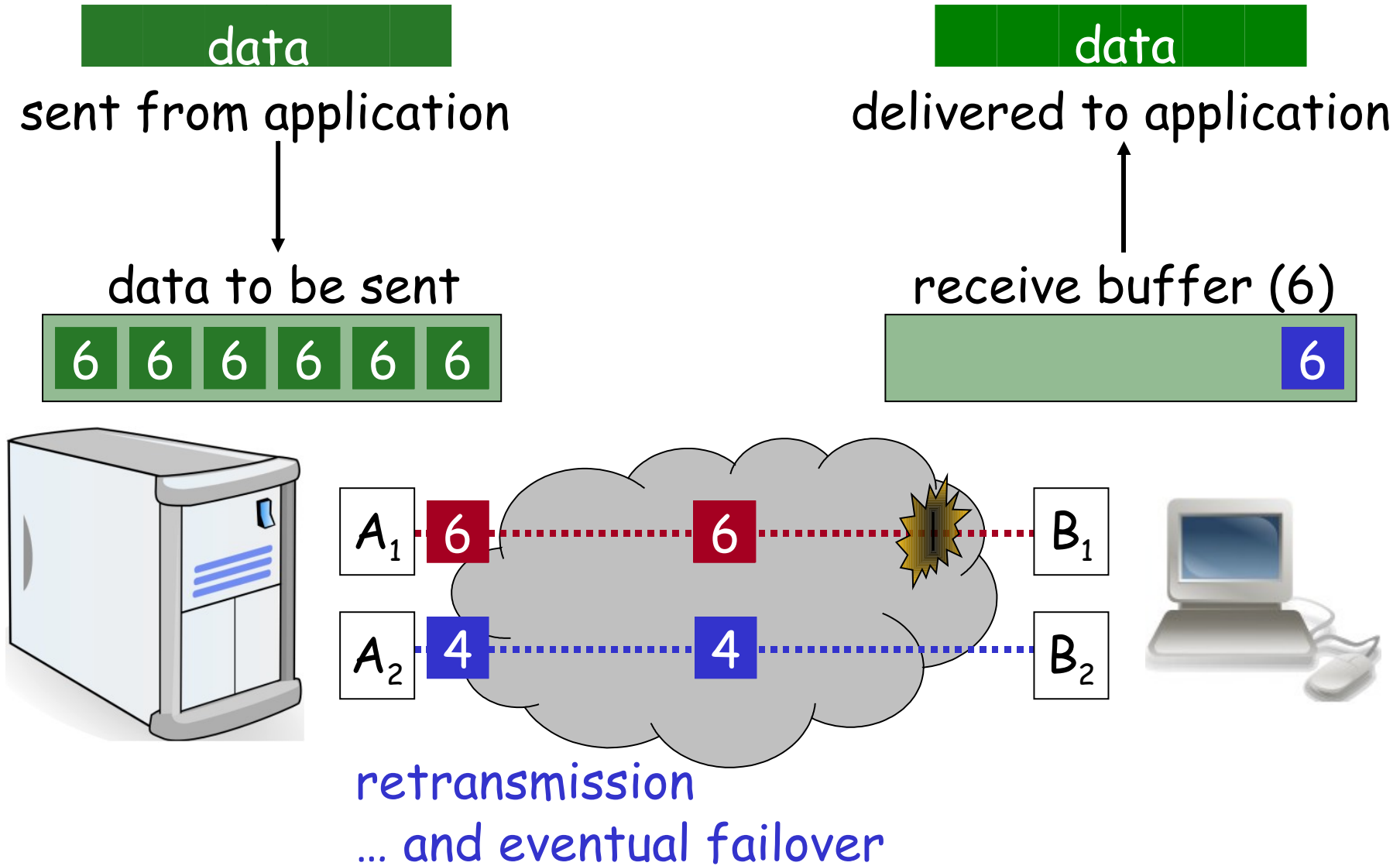
- When a peer is multi-homed, a “primary destination address” will be selected by the SCTP endpoint.
- By default, all data will be sent to this primary address.\*
- When the primary address fails, the sender will select an alternate primary address until it is restored or the user changes the primary address.

\* Concurrent Multipath Transfer (CMT) may change this rule

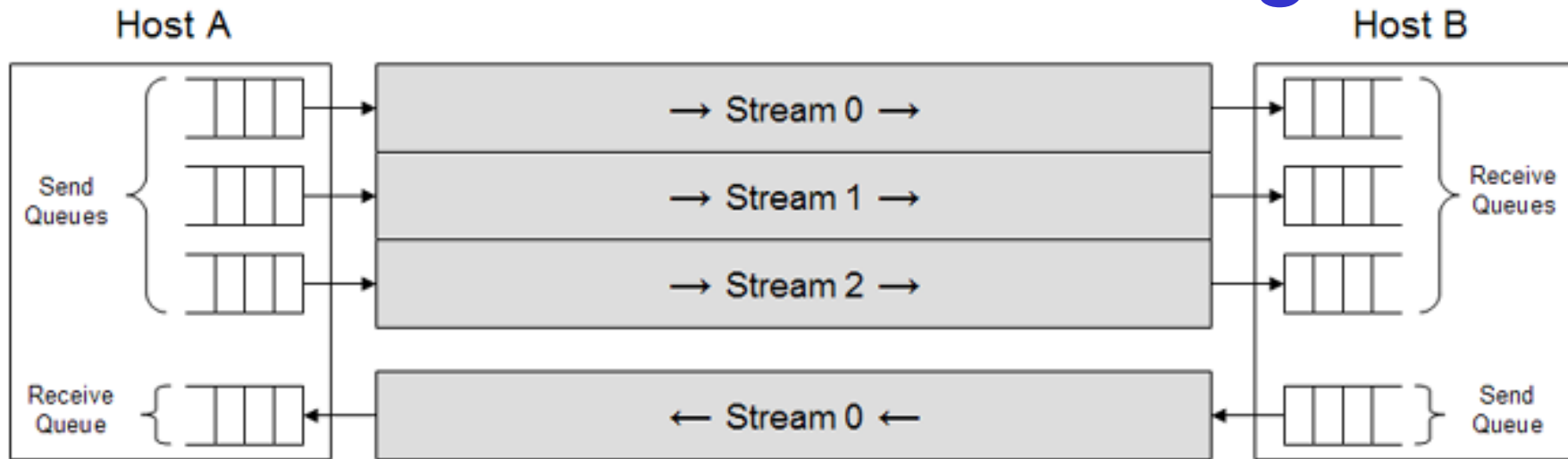
# TCP data transfer with single path failure



# SCTP data transfer with single path failure

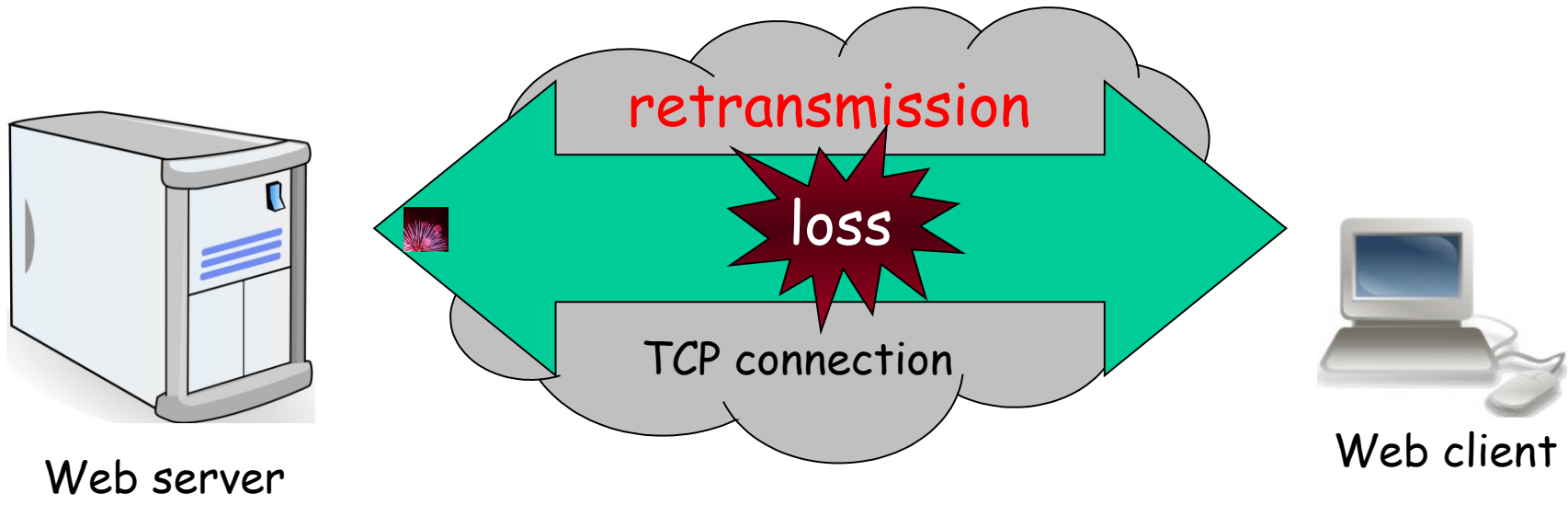
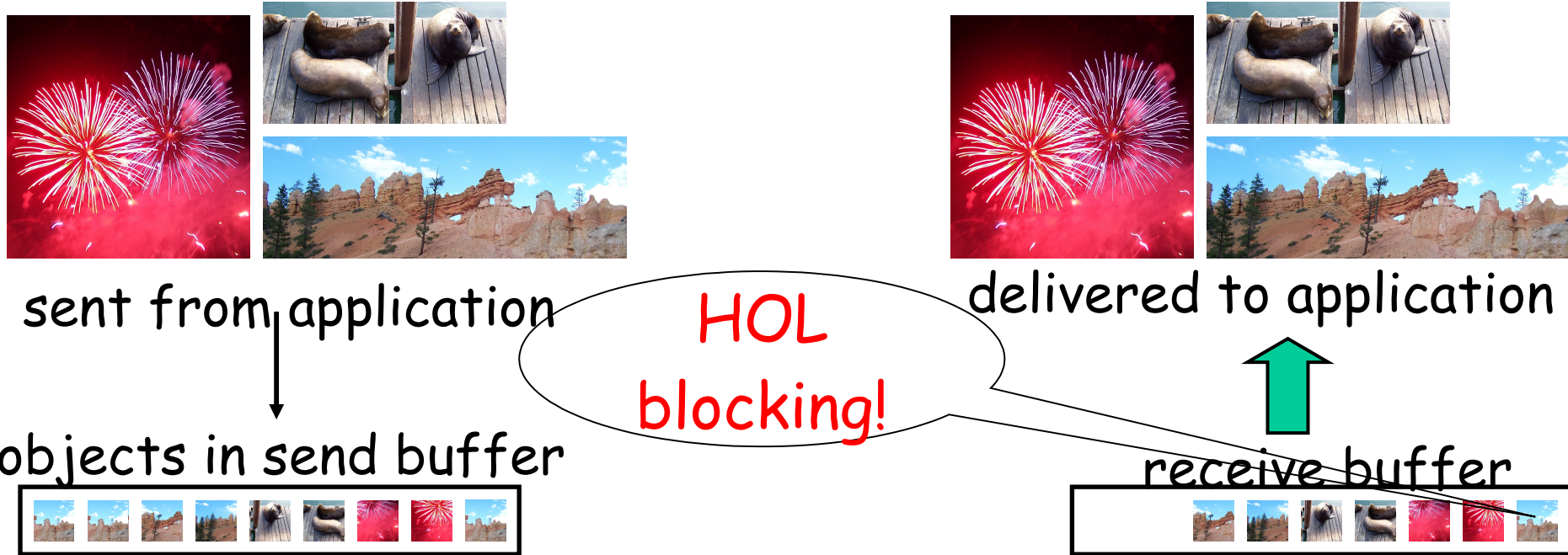


# SCTP Multistreaming

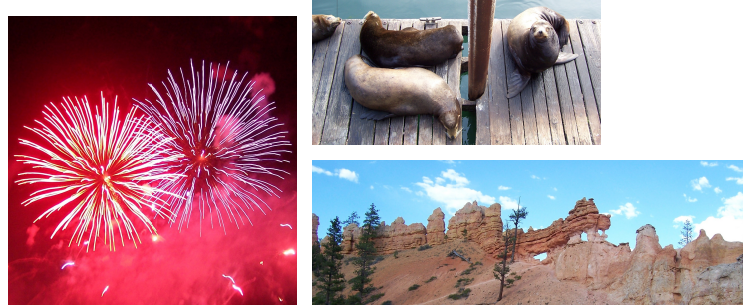


- Logical separation of data within an assoc
- Designed to prevent head-of-line blocking
- Can be used to deliver multiple objects belonging to the same assoc
  - Eg: objects on a webpage, multimedia streams (audio/video/text), files in an FTP *mget*

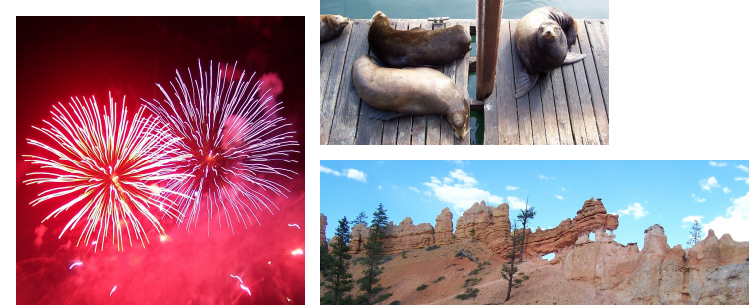
# TCP experiences HOL blocking



# SCTP Multistreaming reduces HOL blocking



sent from application

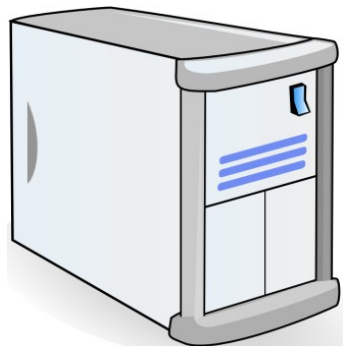


delivered to application

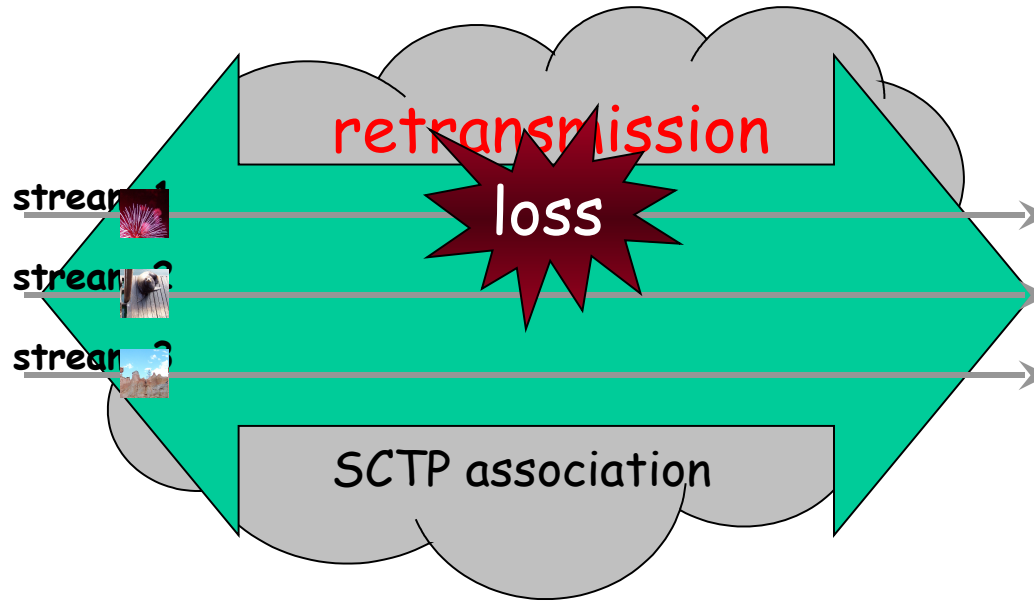
objects in send buffer



receive buffer



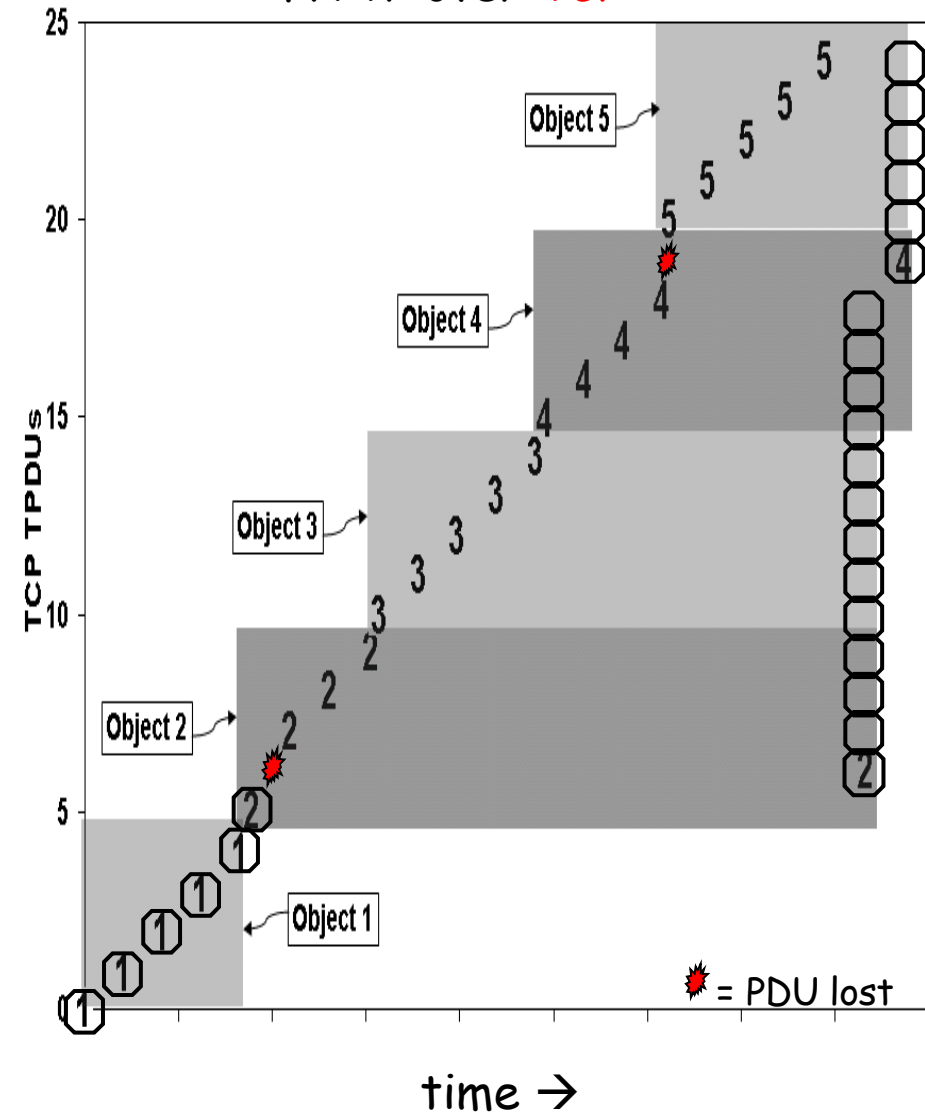
Web server



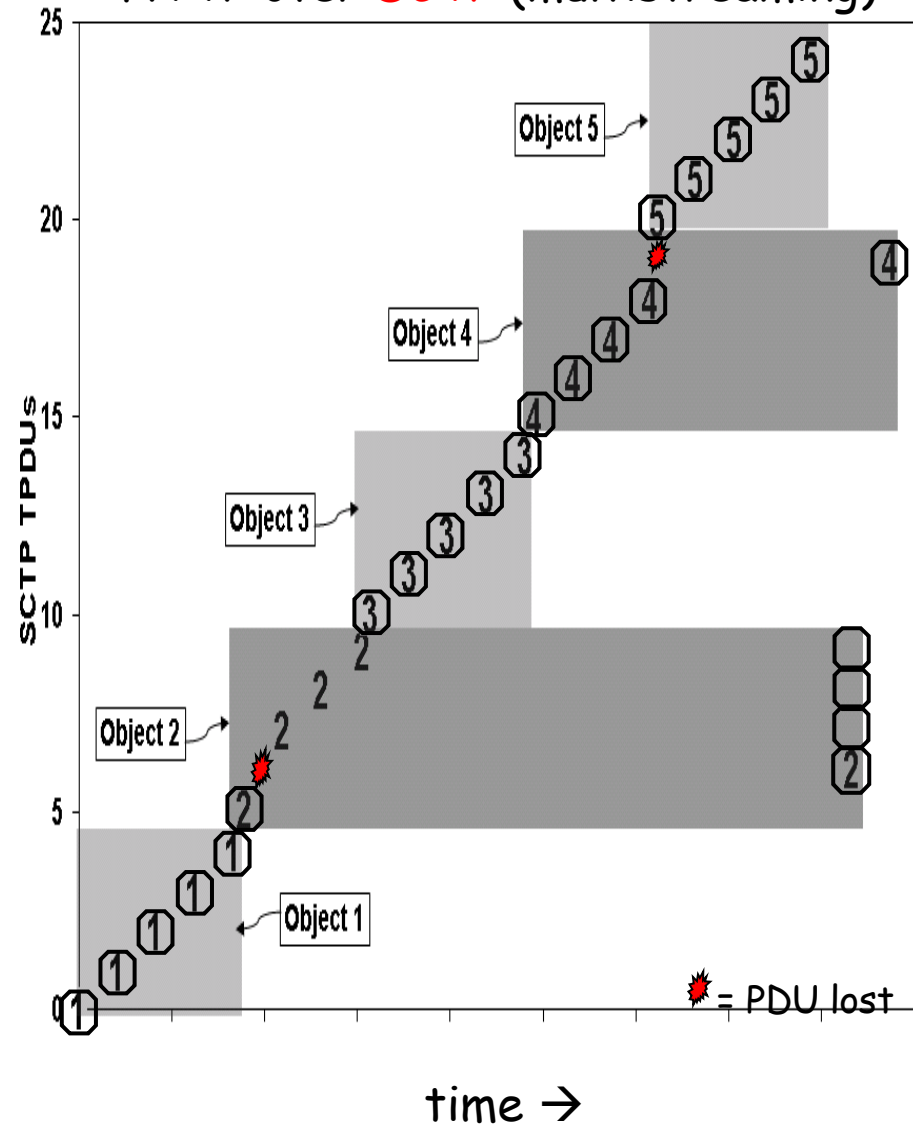
Web client

# Head-of-Line (HOL) Blocking

HTTP over **TCP**



HTTP over **SCTP** (multistreaming)



# TCP work-around to mitigate HOL blocking

- How?
  - Multiple persistent TCP connections to transfer independent web objects
- Problems
  - Possible HOL blocking within one TCP connection
  - No shared sequence space => Less robust to loss detection and recovery
  - Increased load on web server
  - Increased connection establishment latency during SYN losses.
  - Aggressive behavior during congestion

# SCTP and TCP: Similarities

- Both use a handshake to setup and terminate the state (communication) relationship between peers
- Both have an abortive method to terminate the state
- Both provide a “reliable ordered” service:
  - Lost data is retransmitted
  - Data is (or can be) delivered in the order sent
- Both follow an AIMD-based congestion control mechanism.

# SCTP and TCP

- SCTP uses a four-way handshake to setup an *association*. TCP uses a three-way handshake to setup a *connection*.
- However, this does not mean that data can start to be sent more quickly (relative to the start of the connection) with TCP.
- SCTP can exchange data on the third and fourth leg of its handshake. TCP in practicality does not (due to socket API issues).

# SCTP and TCP

- SCTP delivers messages, not a “byte stream”
  - An application using TCP must “frame” its own messages
- SCTP streams allows “partially ordered” transfers
  - Escapes head of line blocking, while preserving order within each stream
- An SCTP sender can send all messages in a single ordered stream to achieve the same behaviour as TCP.

# SCTP and TCP

- SCTP also provides a “reliable un-ordered” service for applications

# Where is it in BSDish systems?

- Available in FreeBSD 7.0 (patch avail for 6.0, 6.1 and 6.2)
- Available as a KLM for MAC O/S X
- Available for NetBSD, But?
- Not supportable on OpenBSD :-)

# Where is it?

- Other O/S's have it too
  - Linux (lk-sctp project),
  - HP-UX (from Emerson),
  - Solaris 10,
  - AIX and a
  - myriad of “purchasable” stacks.
- Also a user space open source stack that can run in windows (supported by Kyoto Univ).

# Other stuff

- One of the MOST active groups in the SCTP community is the WIDE SCTP-wg, if you are a WIDE member please join it :-D
- Later this year (August) Kyoto University will be hosting the 9<sup>th</sup> SCTP inter-op, thank you :-D
- SCTP documents continue to move through the IETF, many of which are implemented in most implementations (BSD stack implements all extensions that I know of :-)